Atlas™ User's Manual

Document no: MD00005
Revision 01.10

October 12, 2000

MIPS Technologies, Inc.
1225 Charleston Road
Mountain View, CA 94043

Notice

Copyright (c) 1999,2000 MIPS Technologies, Inc. All rights reserved.

Unpublished rights reserved under the Copyright Laws of the United States of America.

This document contains information that is proprietary to MIPS Technologies, Inc. ("MIPS Technologies"). Any copying, modifying or use of this information (in whole or in part) which is not expressly permitted in writing by MIPS Technologies or a contractually-authorized third party is strictly prohibited. At a minimum, this information is protected under unfair competition laws and the expression of the information contained herein is protected under federal copyright laws. Violations thereof may result in criminal penalties and fines.

MIPS Technologies or any contractually-authorized third party reserves the right to change the information contained in this document to improve function, design or otherwise. MIPS Technologies does not assume any liability arising out of the application or use of this information. Any license under patent rights or any other intellectual property rights owned by MIPS Technologies or third parties shall be conveyed by MIPS Technologies or any contractually-authorized third party in a separate license agreement between the parties.

The information contained in this document constitutes one or more of the following: commercial computer software, commercial computer software documentation or other commercial items. If the user of this information, or any related documentation of any kind, including related technical data or manuals, is an agency, department, or other entity of the United States government ("Government"), the use, duplication, reproduction, release, modification, disclosure, or transfer of this information, or any related documentation of any kind, is restricted in accordance with Federal Acquisition Regulation 12.212 for civilian agencies and Defense Federal Acquisition Regulation Supplement 227.7202 for military agencies. The use of this information by the Government is further restricted in accordance with the terms of the license agreement(s) and/or applicable contract terms and conditions covering this information from MIPS Technologies or any contractually-authorized third party.

MIPS, R3000, R4000, R5000, R8000 and R10000 are among the registered trademarks of MIPS Technologies, Inc., and R4300, R20K, MIPS16, MIPS32, MIPS64, MIPS-3D, MIPS I, MIPS II, MIPS III, MIPS IV, MIPS V, MDMX, 4K, 4Kc, 4Km, 4Kp, 5K, 5Kc, 20K, 20Kc, EC, MGB, SOC-it, SEAD, YAMON, ATLAS, JALGO, CoreLV and MIPS-based are among the trademarks of MIPS Technologies, Inc.

All other trademarks referred to herein are the property of their respective owners.

Table of contents

Introduction 5
1.1 Purpose of the Atlas platform. 5
1.2 Vital sections 5

2. Getting Started 6
2.1 Supplied hardware 6
2.2 Required hardware 6
2.3 Optional hardware 6
2.4 Wiring it up 6
2.5 Power-up sequence 7

3. Design overview / Block diagram 8

4. Memory map 10
4.1 Devices 11
4.1.1 Revision info 12
4.1.2 Interrupt controller. 12
4.1.3 Hardware acknowledge 13
4.1.4 NMI interrupt controller 14
4.1.5 NMI acknowledge 14
4.1.6 Switches / status 14
4.1.7 Timer 16
4.1.8 Displays 18
4.1.9 Reset control. 20
4.1.10 PSU standby. 21
4.1.11 Flash writing control 22
4.1.12 RTC (real-time clock) 22
4.1.13 UART 23
4.1.14 General purpose I/O 24

5. Board layout 25
5.1 Connectors and Jumpers 27
5.2 Switches 28
5.3 Displays / LEDs 30

6. Hardware description 31
6.1 Compact PCI interface 31
6.2 Primary PCI bus 31
6.3 PCI Arbiter 32
6.4 Power 32
6.5 Reset 32
6.6 Clocks 32
6.7 Interrupt 32
6.8 Serial ports 33
6.9 Ethernet 34
6.10 Video 35
6.11 USB / keyboard / mouse / parallel port 35
6.12 SCSI 35
6.13 Flash memory 35
6.14 EEPROM 36

6.15 JTAG scan chain. 36
6.16 Debug access 36
6.17 Timer/RTC 37

7. SDB Revisions 38

8. 1284 Flash download format 39

9. Core Card design 42
9.1 Interfaces 42
9.1.1 Power 42
9.1.2 PCI Bus 42
9.1.3 Clock 42
9.1.4 Revision number 42
9.1.5 I2C bus 42
9.1.6 Interrupts 42
9.1.7 Endian 42
9.1.8 CBUS 43
9.1.9 EJTAG 44
9.1.10 Misc. 44
9.2 Signals 44
9.2.1 J3 Connector 46
9.2.2 J4 Connector 47
9.3 Physical design 48

10. References 51

11. Revision history 52

1. Introduction

This document is for the user of the Atlas™ platform. It details how to use the board, set it up, and the software-relevant information.

1.1 Purpose of the Atlas platform.

The Atlas platform was designed in order to provide a standard platform for development work with MIPS32™ and MIPS64™ CPUs. It is compatible with the Microsoft® Harp SDB specification (Ref [2]) and as such can be used for Windows® CE development work.

The design is composed of two parts, the Atlas Board, which is 6U CompactPCI form factor and holds the CPU-independent parts of the circuitry, and a “Core” Card, which holds the CPU plus its System Controller and fast SDRAM memory. It is intended for use in a suitable CompactPCI rack system.

1.2 Vital sections

To save you the trouble of looking in the Contents list, the most immediately interesting sections are probably:

- Block diagram (Section 3)
- Memory map (Section 4)
- Hardware jumpers & configuration options (Section 5.1)

It is probably also well worth getting acquainted with the literature in the References section.

2. Getting Started

2.1 Supplied hardware

The basic Atlas hardware consists of:

- Atlas Board.

2.2 Required hardware

In addition to the basic Atlas hardware, you will need:

- Compact PCI rack system or suitable enclosure with ATX power supply.
- The Core Card that carries your choice of CPU.
- Serial cable for RS232 Debug connection. The cable must be a Null Modem cable with 9 way ‘D’ female connector in both ends.

2.3 Optional hardware

The following may also be useful depending on your application:

- Video cable for VGA monitor.
- Ethernet cable.
- USB cable.
- PS2 Keyboard / mouse.
- SCSI cable (Narrow, SCSI-2, 50-pin connector).

In-depth debugging requires the following:

- LA probe connectors (that match with the AMP Mictor headers) if you have a HP Logic Analyser.
- PCI probe board if you want to be able to monitor activity on the internal PCI bus (e.g. FuturePlus FS2000)
- CompactPCI probe board for the backplane bus (e.g. FuturePlus FS3020)
- Parallel port download cable for programming of Flash memory.

2.4 Wiring it up

First, if this is not already fitted, connect the Core Card. It is fairly obvious how to do this, the connectors (J3 and J4) have the same numbers on both boards if you are in doubt about the orientation. One of the corner mounting pillars is also offset to prevent incorrect insertion.

When removing the Core Card - **be careful!** There will typically be BGA devices fitted which can take offence at having the board bent violently. Under each corner of the Core Card is a mounting pillar, with a gap where a screwdriver can be inserted to gently lever it up. Only apply the screwdriver to the PCB area around the mounting holes - you don’t want to cut any tracks by accident.

You will most likely want to also have the following set up:

- Serial port. The supplied PROM monitor (YAMON) by default signs on via the serial debug port, using 38.4 kbaud, 8 bits/char, RTS/CTS hardware handshaking and no parity. A 5-wire cable is sufficient. The implemented signals must be RXD, TXD, RTS, CTS and GND, see Section 6.8 , “Serial ports” for serial connector pinout.
- Ethernet. Twisted -pair ethernet cable will plug into the front panel, this will run at 10 Mbit/s, half-duplex.
- Check that the 4-way DIP switches S1 and S5 are in the positions labelled as default in Table 32.

2.5 Power-up sequence

If you are using an ATX PSU via J12, when you first connect the power supply and switch it on the only thing that happens is that the green “5VSB” LED turns on because the ATX power supply comes up by default into “standby” mode. Press the switch marked “ON/NMI” (S4) to bring up the board. Check that the green “3V3” and “5V” LEDs also turn on to indicate good power.

NOTE: some ATX power supplies are unable to maintain power when the load is light . If you experience problems where the “3V3” or “5V” LED turns off without cause or never turns on then check the power supply. A dummy load can be added to 5V to solve the problem but the best solution is to get a good power supply

If you on the other hand are drawing power via the CompactPCI connectors then the board will boot as soon as there is power to the backplane.

The green “FPGA” LED should be on - this indicates that the board’s FPGA has booted.

The red “RST” LED should be off - if lit, this indicates that something is holding the board in reset.

When the CPU initially boots, the YAMON monitor signs on using the Dbg Ser serial port (the upper one) with information about the board configuration details, i.e. board revision, SDRAM size etc.

Finally you should arrive at YAMON’s prompt line. Simultaneously, you should see the word “YAMON” on the ASCII LED display. If you do not see this, check with the YAMON User’s Manual for what the display messages mean.

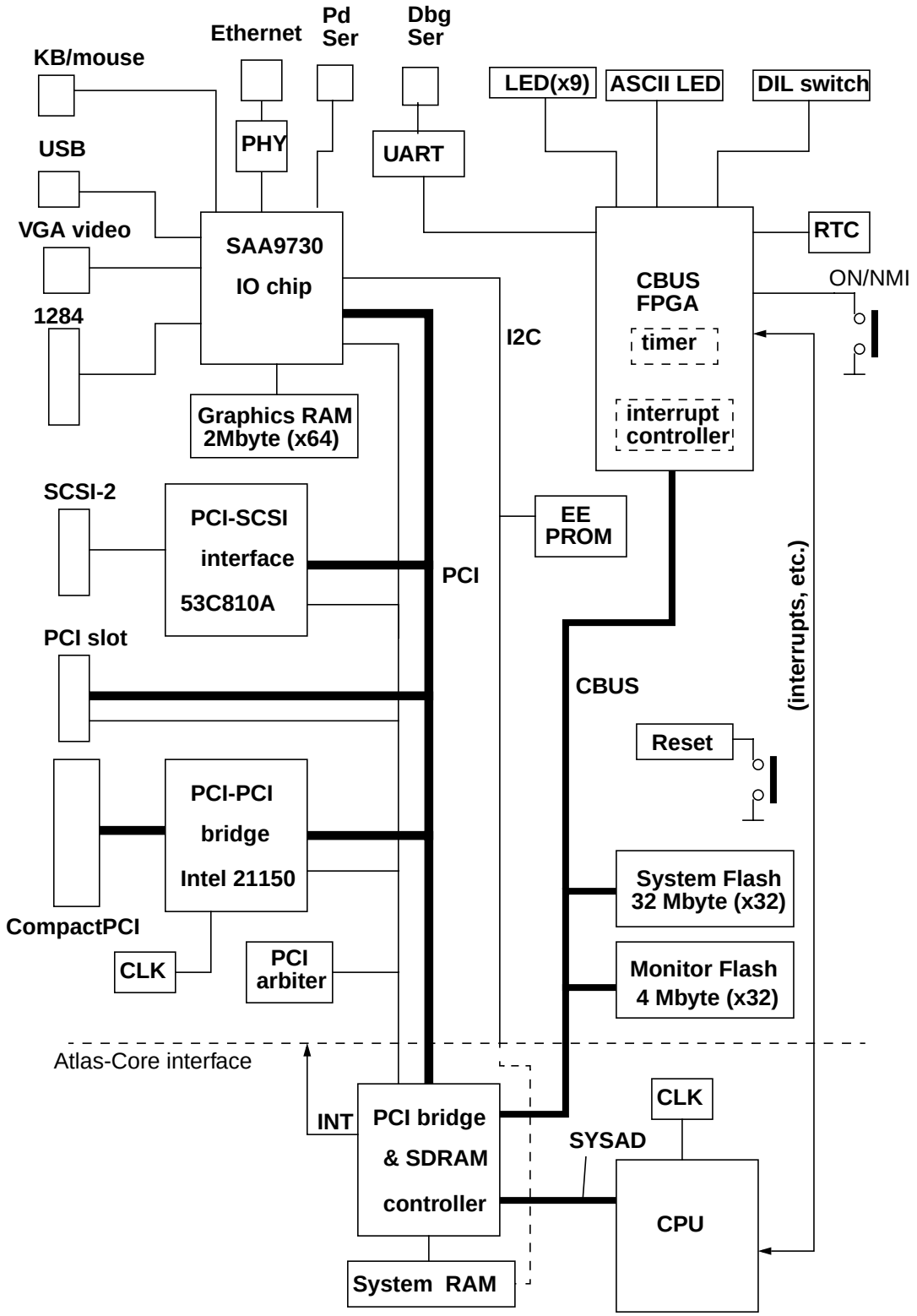
Note that you will be running in big-endian mode, and that you can change this with a switch setting. See the Atlas User’s Manual for details.

The command “help” lists the available commands in the YAMON monitor, and “help <command name>” gives more detailed information about the specific command. See the YAMON documentation for a full description of the functionality.

3. Design overview / Block diagram

The diagram below gives an overview of the important features of the Atlas Board.

Figure 1. Block diagram



The Core Card design shown is only a typical implementation (in fact, most implementations will look like this until CPUs that have outlived the SysAD bus start arriving). However all Core Cards will conform to the same interface specification.

Worth noting is that the Core Card has its own clock. Nothing on Atlas is synchronized to this clock, as it is (most probably) independent of the PCI clock, and the CBUS protocol is asynchronous by nature.

The CBUS exists to allow the CPU to access peripherals which either have to be available before the PCI bus has been configured (for example the Flash memory it's booting from) or those that require simple, low-latency access, for example the interrupt controller, the debug LEDs and so on.

The Flash is divided up into two areas, known as "Monitor" and "System". In principle, the Monitor area is inviolate - it is there so you can easily recover if you should happen to corrupt the System Flash entirely. It can be reprogrammed but only via a hardware download cable. The System area can be reprogrammed at will. Atlas can be configured by setting S1-1 to boot from either Flash area.

The primary PCI bus is 32-bit, 33 MHz PCI standard version 2.1 compliant. It connects to the backplane CompactPCI bus via the bridge chip, and allows devices on the bus (and backplane) DMA access to the RAM on the Core Card. A single 5V PCI slot is provided on the Atlas to allow insertion of optional peripherals, and also provides a way of monitoring traffic on this bus.

On the primary PCI bus are:

- Philips SAA9730 IO controller. This device provides a large number of interface functions, the "Pd Ser" serial port, Ethernet, USB, IEEE1284, keyboard and mouse, plus it's a powerful graphics controller. Additionally, it provides a I2C serial bus, which gives access to the EEPROM containing non-volatile data and any similar devices on the Core Card (e.g. SDRAM modules).
- LSI Logic (used to be Symbios) SYM53C810A SCSI controller. Self explanatory.
- Intel 21150 PCI-PCI bridge chip. Self explanatory.
- A 5V PCI board slot.
- The Core Card.

4. Memory map

The memory map as seen from the CPU is partly dependent on the Core Card, and how the PCI bus is configured. However the Atlas design places some requirements on how the CBUS is mapped, and therefore this area will always appear as shown below.

Table 1. Atlas physical memory map.

Base address	Size	Function
0000.0000	128Mbytes	Typically SDRAM (on Core Card)
0800.0000	256 Mbytes	Typically PCI
1800.0000	62 Mbytes	Typically PCI
1BE0.0000	2 Mbyte	Typically System Controller specific
1C00.0000	32 Mbytes	System Flash
1E00.0000	4 Mbytes	Monitor Flash
1E40.0000	12 Mbytes	reserved
1F00.0000	1 Mbyte	Interrupt controller
		Timer
		Switches
		LEDs
		ASCII display
		Soft reset
		FPGA revision number
		RTC
		UART
1F10.0000	11 Mbytes	Typically System Controller specific.
1FC0.0000	1 Mbyte	S1-1 ON: Base of Monitor Flash (0x1E00.0000) S1-1 OFF: At base of upper 4 Mbyte of System Flash (0x1DC0.0000)
1FD0.0000	3 Mbytes	Typically System Controller specific.

Note: The shaded area of the table indicates memory areas the mapping of which depends on the implementation of the Core Card, and of software. The indicated mappings shown would be suitable for a Harp SDB implementation.

Note: Which physical Flash that is being programmed when programming address range 1FC0.0000 is dependent of the position of switch S1-1.

Note: Address 1FC0.0010 is “special”, in the sense that when the software reads this address it is overridden and does NOT decode to an address in Flash, but rather to register address REVISION. This is done to ensure future compatibility - all possible implementations of MIPS Technologies boards will be able to identify their hardware environment and configure themselves accordingly. When programming address 1FC0.0000 it is the Flash selected by switch S1-1 that is being programmed, and the only way to read the programmed value back is to read the dedicated System or Monitor Flash address. Reads from address 1E00.0010 will decode to an address in Flash.

RAM is typically mapped at the bottom of memory so that the exception vectors are located in fast memory.

Atlas does not specify a mapping for addresses above 0x2000.0000 which cannot be addressed in kseg0/1.

See the appropriate data sheets for the 21150AB, the SAA9730 and the SYM53C810A PCI devices to see how to configure these. The I2C bus is controlled by the SAA9730, its slave address map is configured as follows:

Table 2. I2C slave address map for EEPROM devices.

I2C slave address	Size	Function
0x50	256 bytes	Core Card, PC-100 SDRAM
0x51	256 bytes	Core Card, optional PC-100 SDRAM
0x52	256 bytes	Core Card, optional PC-100 SDRAM
0x53	256 bytes	Core Card, optional PC-100 SDRAM
0x54 - 0x55	512 bytes	Atlas EEPROM, unused area.
0x56 - 0x57	512 bytes	Atlas EEPROM, contains MAC address and serial number.

4.1 Devices

Devices in the memory map above are listed below, with their internal register maps and programming details. Note that all addresses listed below are physical addresses. It is recommended to use the macros available in the YAMON source deliverable (See Ref[6]) for accessing these registers and bitfields.

All registers shall be addressed as 32-bit words, and lie on 64-bit word boundaries. This convention allows software to access all registers using the same word address in both little and in big endian mode.

Those registers that contain a single value are not described in bit-field detail, these values are aligned to the LSB of the word.

4.1.1 Revision info

Name: REVISION
Address: 0x1FC0.0010
Access: RO
Reset value: n/a

Table 3. REVISION register

Bits	Field name	Function	Initial value
31:24	reserved		0
23:16	FPGRV	8 bit binary number gives revision of CBUS FPGA.	n/a
15:10	CORID	6 bit Core Card ID	n/a
9:8	CORRV	2 bit Core Card revision	n/a
7:4	PROID	4 bit binary number gives product ID	0x0
3:0	PRORV	4 bit binary number gives product revision.	n/a

A cryptic word can be read here, which imparts information about the revision of the Atlas and Core Cards. This information is for MIPS Technologies internal use.

4.1.2 Interrupt controller.

All sources of interrupt except NMI on the Atlas / Core card plus those from the CompactPCI bus are handled in an interrupt controller located in the CBUS FPGA. They are combined together to generate an interrupt on the Core Cards INTN[0] signal. The interrupt signals INTN[5:1] are always inactive. It is recommended to use the macros available in the YAMON source deliverable (See Ref[6]) for accessing these registers.

The various interrupt sources are listed below together with their bit-positions in the interrupt controller registers.

NOTE: The field bits may not correspond to the actual signal voltage, as many of these are active low. All the interrupt bits are active high sense.

Table 4. Interrupt controllers registers field-positions

Bits	Field name	Function
31:20	reserved	
19	SERR	Primary PCI SERR# signal
18	INTD	Primary PCI INTD# (PCI slot only)
17	INTC	Primary PCI INTC# (includes SCSI)
16	INTB	Primary PCI INTB# (includes SAA9730)
15	INTA	Primary PCI INTA# (PCI slot only)
14	ATXFAIL	Indicates ATX PSU about to fail
13	DEG	The DEG# signal from the PCI backplane - indicates power supply is about to fail
12	ENUM	The ENUM# signal from the PCI backplane
11	PCID	Compact PCI INTD#
10	PCIC	Compact PCI INTC#
9	PCIB	Compact PCI INTB#

Table 4. Interrupt controllers registers field-positions

Bits	Field name	Function
8	PCIA	Compact PCI INTA#
7	reserved	
6	CORELO	Core Card low priority interrupt
5	COREHI	Core Card high priority interrupt
4	RTC	Real Time Clock
3:2	reserved	
1	TIM0	Timer 0 (in CBUS FPGA)
0	SER	Auxiliary serial port chip

The interrupt controller provides the following registers to enable control of interrupts:

Table 5. Interrupt Controller Registers. BASE = 0x1F00.0000

Name	Offset Address	Access	Function	Initial value
INTRAW	0x0000.0000	RO	"Raw" value on external interrupt lines. NOTE: This may not correspond to the actual signal voltage, as many of these are active low. All these INTRAW bits are active high sense.	n/a
INTSETEN	0x0000.0008	WO	Writing a "1" in a given bit position sets the corresponding bit in the INTENABLE register.	0
INTRSTEN	0x0000.0010	WO	Writing a "1" in a given bit position clears the corresponding bit in the INTENABLE register.	0xFFFF.FFFF
INTENABLE	0x0000.0018	RO	Each bit that is a "1" indicates that the corresponding interrupt is enabled.	0
INTSTATUS	0x0000.0020	RO	Shows the status of enabled interrupts only, i.e. INTRAW anded with INTENABLE.	0

Note that several interrupts can be enabled or disabled in the same operation, but that it is impossible to both enable and disable interrupts in the same operation.

4.1.3 Hardware acknowledge

The DEG interrupt is by nature transient. Therefore it is debounced and latched and thereafter treated as ordinary level-based interrupt in the interrupt controller. It can be cleared by writing to HWINTACK. When the bit in this register is written as "1", the interrupt is cleared.

Name: HWINTACK
Address: 0x1F00.0100
Access: WO
Reset value: n/a

Table 6. HWINTACK register

Bits	Field name	Function	Initial value
31:14	reserved		n/a
13	DEG	Write 1 to acknowledge DEG	n/a
12:0	reserved		n/a

4.1.4 NMI interrupt controller

When the ON/NMI push button is activated the signal is debounced and latched in the NMI interrupt controller. This signal then generates an interrupt on the MIPS core LV NMIN pin.

Name: NMISTATUS
Address: 0x1F00.0024
Access: RO
Reset value: n/a

Table 7. NMISTATUS register

Bits	Field name	Function	Initial value
31:1	reserved		n/a
0	ONNMI	Pending NMI from ON/NMI push button	n/a

4.1.5 NMI acknowledge

The ON/NMI interrupt is by nature transient. Therefore it is debounced & latched and thereafter treated as ordinary level-based interrupt in the NMI interrupt controller. The NMI interrupt can be cleared by writing to NMIACK. When the bit in this register is written as “1”, the interrupt is cleared.

Name: NMIACK
Address: 0x1F00.0104
Access: WO
Reset value: n/a

Table 8. NMIACK register

Bits	Field name	Function	Initial value
31:1	reserved		n/a
0	ONNMI	Write 1 to acknowledge NMI	n/a

4.1.6 Switches / status

The following registers allow software to monitor the state of various switches and jumpers on the Atlas board. All DIP switches give a value of “1” for a switch in the “ON” or “CLOSED” position.

There is no debouncing on these registers, so if software wants to monitor a value while it changes, allowance for this must be made by waiting for the new value to become stable.

Name: SWITCH
Address: 0x1F00.0200
Access: RO
Reset value: n/a

Table 9. SWITCH register

Bits	Field name	Function	Initial value
31:12	reserved		0
11	S14	DIP switch S1 bit 4.	n/a
10	S13	DIP switch S1 bit 3. Flash write protection.	n/a
9	S12	DIP switch S1 bit 2.	n/a
8	S11	DIP switch S1 bit 1. Flash address mapping.	n/a
7:0	S2	8-bit value of the setting of DIP switch S2. LSB is set by the lefthand switch.	n/a

See Section 5.2 for description of switch S1 functionality.

Name: STATUS
Address: 0x1F00.0208
Access: RO
Reset value: n/a

Table 10. STATUS register

Bits	Field name	Function	Initial value
31:5	reserved		0
4	MFWR	“0” indicates Monitor Flash lock bits are write enabled (JP8 fitted).	n/a
3	S54	“1” will set YAMON in factory default mode (communication on serial debug port etc.) as controlled by switch S5-4.	n/a
2	S53	DIP switch S5-3	n/a
1	BIGEND	“1” indicates big endian mode as controlled by switch S5-2.	n/a
0	PCISYS	“1” indicates that Atlas is plugged in to a System Slot of a Compact PCI backplane. (SYSENN signal).	n/a

Name: JMPRS
Address: 0x1F00.0210
Access: RO
Reset value: n/a

Table 11. JMPRS register

Bits	Field name	Function	Initial value
31:2	reserved		0
1	EELOCK	State of JP4: Not fitted ~ “1” = EEPROM write protected.	n/a
0	PCI33M	State of JP3: (PCI freq). 1= 33MHz, 0= 16.5MHz.	n/a

4.1.7 Timer.

To allow accurate timing measurement a fast timer is implemented. It run from the fixed 40 MHz clock associated with the SCSI controller, so it is independent of CPU and bus speeds. It generate an interrupt when the count value match a compare value. This interrupt is latched until cleared by writing to TMINTACK.

The timer device provides the following registers to control of the timer resources:

Table 12. Timer Registers. BASE = 0x1F00.0300

Name	Offset Address	Access	Function
TM0CNT	0x0000.0000	R/W	Timer 0 current count value. When written, value is updated. The count value are reset to zero when the compare value (TM0CMP) is updated.
TM0CMP	0x0000.0008	R/W	Timer 0 current compare value. When the counter reaches this value, it resets TM0CNT to zero and continues counting, and an interrupt signal (TIM0) is generated. This is a momentary event, so the interrupt latches and must be acknowledged by writing to the TMINTACK register. When the TM0CMP value is updated the TM0CNT is reset to zero.
TMINTACK	0x0000.0020	WO	Bit 0: Write 1 to clear interrupt TIM0

Name: TM0CNT
Address: 0x1F00.0300
Access: R/W
Reset value: 0x0000.0000

Table 13. TM0CNT register

Bits	Field name	Function	Initial value
31:24	reserved		0x00
23:0	COUNT	Timer 0 current count value. When written, value is updated.	0x00.0000

Name: TM0CMP
Address: 0x1F00.0308
Access: R/W
Reset value: 0x0000.0000

Table 14. TM0CMP register

Bits	Field name	Function	Initial value
31:24	reserved		0x00
23:0	COMP	Timer 0 current compare value. When the counter reaches this value, it resets to zero and continues counting, and an interrupt signal (TIM0) is generated. This is a momentary event, so the interrupt latches and must be acknowledged by writing to the TMINTACK register.	0x00.0000

Name: TMINTACK
Address: 0x1F00.0320
Access: WO
Reset value: n/a

Table 15. TMINTACK register

Bits	Field name	Function	Initial value
31:1	reserved		n/a
0	TM0ACK	Write “1” to acknowledge interrupt of TIM0	n/a

4.1.8 Displays

There are 3 display devices on the front panel: The single green LED, an 8-LED array, and an 8-character ASCII display (HDSP-2532). These are controlled through the following registers.

Table 16. Display Registers. BASE = 0x1F00.0400

Name	Offset Address	Access	Function
LEDGREEN	0x0000.0000	R/W	Bit 0: Green LED is ON.
LEDBAR	0x0000.0008	R/W	8 bits each corresponding to 1 LED (1 = ON).
ASCIIWORD	0x0000.0010	WO	Writing a 32-bit word to this register will cause it to be displayed in hex on the ASCII character display.
ASCIIPOS0	0x0000.0018	WO	Writing an ASCII value to this register updates ASCII display position '0', which is the left-most positioned character.
ASCIIPOS1	0x0000.0020	WO	Writing an ASCII value to this register updates ASCII display position '1'.
ASCIIPOS2	0x0000.0028	WO	Writing an ASCII value to this register updates ASCII display position '2'.
ASCIIPOS3	0x0000.0030	WO	Writing an ASCII value to this register updates ASCII display position '3'.
ASCIIPOS4	0x0000.0038	WO	Writing an ASCII value to this register updates ASCII display position '4'.
ASCIIPOS5	0x0000.0040	WO	Writing an ASCII value to this register updates ASCII display position '5'.
ASCIIPOS6	0x0000.0048	WO	Writing an ASCII value to this register updates ASCII display position '6'.
ASCIIPOS7	0x0000.0050	WO	Writing an ASCII value to this register updates ASCII display position '7', which is the right-most positioned character.

Name: LEDGREEN
Address: 0x1F00.0400
Access: R/W
Reset value: 0x01

Table 17. LEDGREEN register

Bits	Field name	Function	Initial value
31:1	reserved		0
0	ON	Set GREEN LED ON (=1)	1

Name: LEDBAR
Address: 0x1F00.0408
Access: R/W
Reset value: 0x00

Table 18. LEDBAR register

Bits	Field name	Function	Initial value
31:8	reserved		0
7:0	BAR	8 bits each corresponding to 1 LED (1 = ON).	0x00

Name: ASCIIWORD
Address: 0x1F00.0410
Access: WO
Reset value: n/a

Table 19. ASCIIWORD register

Bits	Field name	Function	Initial value
31:0	HEX	Writing a 32-bit word to this register will cause it to be displayed in hex on the ASCII character display.	n/a

Name: ASCIIPOS0
Address: 0x1F00.0418
Access: WO
Reset value: n/a

Table 20. ASCIIPOS0 register

Bits	Field name	Function	Initial value
31:8	reserved		n/a
7:0	ASCII	Writing an ASCII value to this register updates ASCII display position ‘0’. Position ‘0’ is the left-most positioned character.	n/a

Register descriptions for ASCIIPOS1 to ASCIIPOS7 are identical with the description above for ASCIIPOS0.

4.1.9 Reset control.

There are two different reset functionalities that are controlled by software. Writing a “magic” value to the SOFTRES register immediately triggers a reset of the Atlas Board, and the BRKRES register controls how the “break” condition on the Debug Serial port is monitored. Both reset functions generate a board reset with the exact same effect as if you had pressed the reset button.

Name: SOFTRES
Address: 0x1F00.0500
Access: WO
Reset value: 0x00

Table 21. SOFTRES register

Bits	Field name	Function	Initial value
31:8	reserved		n/a
7:0	RESET	Writing the magic value GORESET (==0x42) to this field will initiate a board reset	0x00

Name: BRKRES
Address: 0x1F00.0508
Access: R/W
Reset value: 0x00

Table 22. BRKRES register

Bits	Field name	Function	Initial value
31:8	reserved		0
7:0	WIDTH	Writing a value to this address indicates the number of milliseconds in length a “Break” must be on the “Dbg Ser” interface in order to trigger a reset. Valid values are from 0 to 255. A value of zero prevents this reset ever occurring.	0x0A (i.e. 10ms)

NOTE: The initial value for WIDTH of 10 ms will cause problems, if the baud rate of the serial port is less than 2400 Baud. If baud rates below 2400 Baud are used, this register must be programmed to a larger value.

4.1.10 PSU standby.

If (and only if) you are running from an ATX PSU, you have the ability to shut down into a standby mode until the ON/NMI button is pressed again.

Name: PSUSTBY
Address: 0x1F00.0600
Access: WO
Reset value: n/a

Table 23. PSUSTBY register

Bits	Field name	Function	Initial value
31:8	reserved		n/a
7:0	STBY	Writing the magic value GOSTBY (== 0x4d) to this address will shut down the ATX PSU.	n/a

4.1.11 Flash writing control

The following register provides access to enable for writing (programming) the System Flash devices:

Name: SFWCTRL
Address: 0x1F00.0700
Access: R/W
Reset value: 0x00

Table 24. SFWCTRL register

Bits	Field name	Function	Initial value
31:8	reserved		0
7:0	WRENA	Writing the magic value ENSFWRITE (== 0xC7) to this address will enable writing to the System Flash. It will not however override switch S1-3.	0x00

4.1.12 RTC (real-time clock)

For details on programming this device, see the appropriate data sheet, which is available from their web site (Ref [8]). The device is the Dallas DS1687-5 (with internal battery).

Name: RTCADR
Address: 0x1F00.0800
Access: WO
Reset value: n/a

Table 25. RTCADR register

Bits	Field name	Function	Initial value
31:8	reserved		n/a
7:0	ADR	Writing to this address sets the RTC internal register address, which will be used on subsequent accesses to RTCDAT.	n/a

Name: RTCDAT
Address: 0x1F00.0808
Access: R/W
Reset value: n/a

Table 26. RTCDAT register

Bits	Field name	Function	Initial value
31:8	reserved		0
7:0	DATA	Once RTCADR has been loaded with the RTC internal register address, this pseudo-register can be used to read and write the addressed register.	n/a

4.1.13 UART

For details on programming the UART (a TI 16C550C), see the appropriate data sheet on their web site (Ref [9]). The clock frequency for baud rate calculations is 3.6864 MHz.

The registers of the UART, which have a native width of 8-bit, are all memory mapped on 64-bit aligned boundaries with the following layout:

Table 27. UART Registers. BASE = 0x1F00.0900

Name	Offset Address	Access	Function
RXTX	0x0000.0000	R/W	Receive / Transmit char register
INTEN	0x0000.0008	R/W	Interrupt enable register
IIFIFO	0x0000.0010	R/W	Read: Interrupt identification Write: FIFO control
LCTRL	0x0000.0018	R/W	Line control register[1]
MCTRL	0x0000.0020	R/W	Modem control register
LSTAT	0x0000.0028	R/W	Line status register
MSTAT	0x0000.0030	R/W	Modem status register
SCRATCH	0x0000.0038	R/W	Scratch register

1. The Divisor Latch Registers are accessible through RXTX and INTEN registers when bit 7 (Divisor Latch Access Bit) of the Line Control Register is set.

4.1.14 General purpose I/O

Eight GP inputs and eight GP outputs connected to the Core Card are implemented on the Atlas Board. Read documentation for Core Card for the usage.

Name: GPOUT
Address: 0x1F00.0A00
Access: R/W
Reset value: n/a

Table 28. GPOUT register

Bits	Field name	Function	Initial value
31:8	reserved		n/a
7:0	OUTVAL	Writing to this address sets the 8 GP output pins. Reading gives the actual setting of the GP output pins. Functionality is Core Card dependent.	0x00

Name: GPINP
Address: 0x1F00.0A08
Access: RO
Reset value: n/a

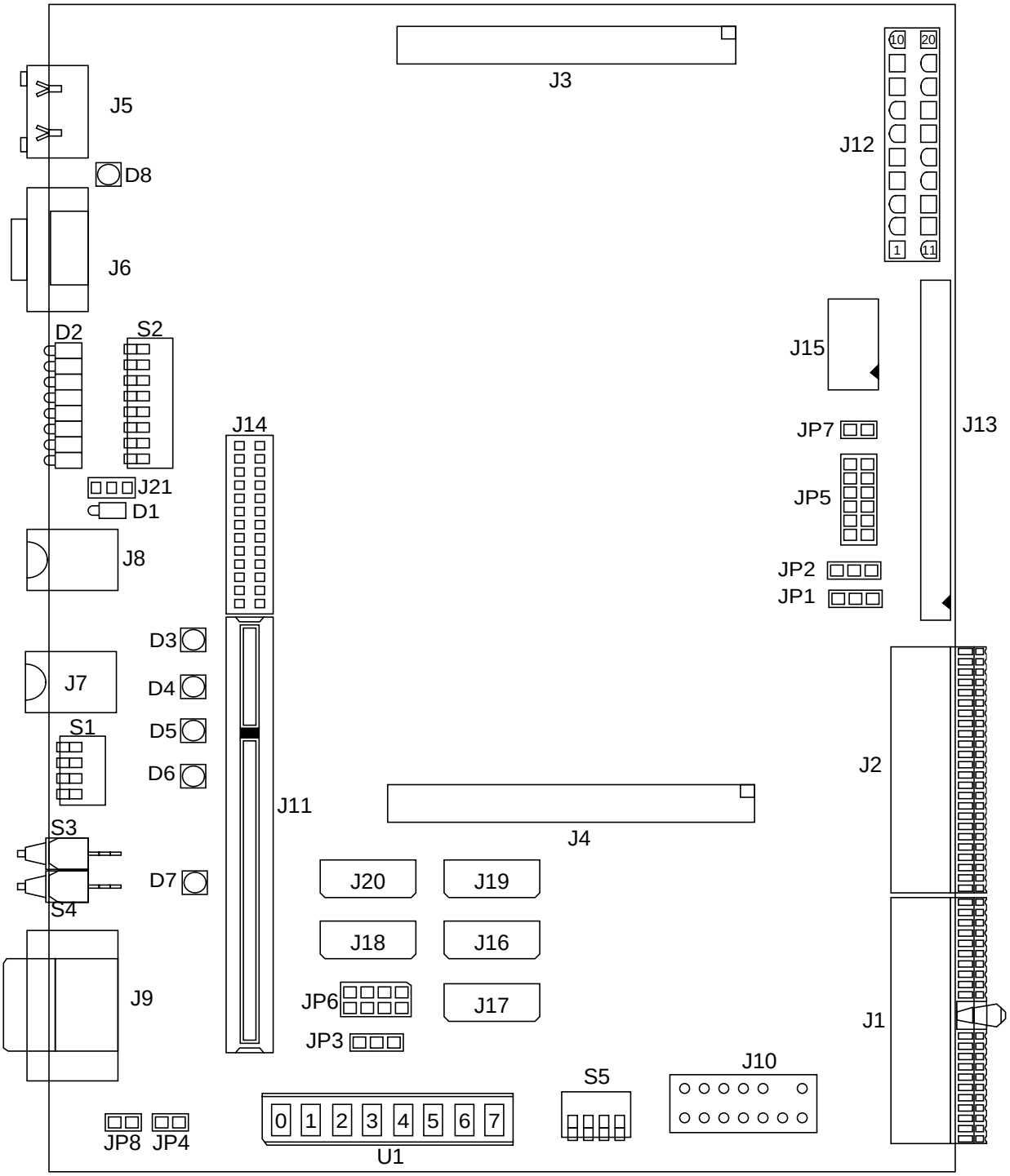
Table 29. GPINP register

Bits	Field name	Function	Initial value
31:8	reserved		0
7:0	INPVAL	Reading gives the actual state of the GP input pins. Functionality is Core Card dependent.	n/a

5. Board layout

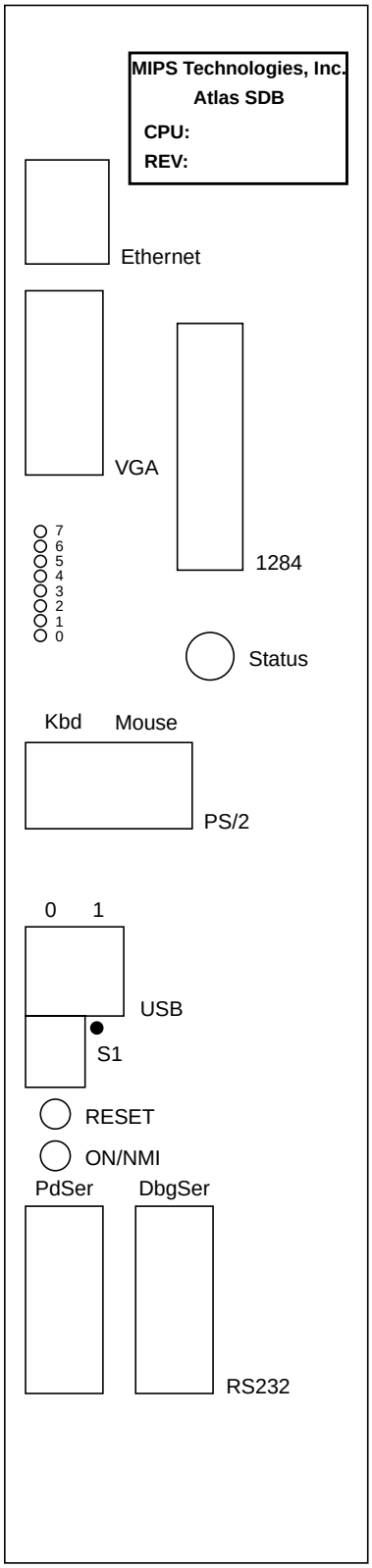
The diagram in Figure 2 shows the basic layout of the Atlas Board.

Figure 2. Atlas board layout



The layout of the front panel (if fitted) is as shown below.

Figure 3. Front panel layout



5.1 Connectors and Jumpers

All jumpers and connectors are listed below. All jumpers are standard 0.1” pitch. It is worth getting some spares in case you lose any. A dot by a pin indicates which pin is pin 1. On all jumpers, pin numbering goes crosswise, i.e. the end pins are 1 and 2, this is not always the case on other connectors.

Table 30. Interface connectors

Ref	Type	Description
J1	CompactPCI	Carries “conventional” PCI signals to/from the backplane.
J2	CompactPCI	Carries the extra CompactPCI signals required of a “System Board” to/from the backplane.
J3	200way header	Connects the Core Card. Carries amongst other things the CBUS.
J4	200way header	Connects the Core Card. Carries amongst other things the primary PCI bus.
J5	RJ45	Ethernet
J6	15 pin DSUB	VGA video connector. Connects to a standard PC monitor.
J7	Dual USB	Two USB host ports.
J8	Dual 6 pin miniDIN	Connection to a PC keyboard (lower) and mouse (upper).
J9	Dual 9 pin DSUB	Serial ports. “Dbg Ser” (upper) and “Pd Ser” (lower).
J10	14 pin 0.1” header	EJTAG connector. As per EJTAG specification, see Ref [7]. Pin 12 removed for keying.
J11	PCI slot	Connects to the primary PCI bus. Allows insertion of probe board, or PCI board with additional functionality.
J12	ATX PSU	This connects the power in a benchtop environment.
J13	50 pin 0.1” header	SCSI interface. Connects to e.g. HDD via IDC ribbon connectors.
J14	26 pin 0.1” header	IEEE1284 / Flash programming port. This can either be used as a parallel port for communications, and is also used to program Flash devices under production.
J15	10 pin 0.1” header	Download connector for Arbiter EPLD.
J16	AMP 38 pin Mictor	HP Logic Analyzer connector
J17	AMP 38 pin Mictor	HP Logic Analyzer connector
J18	AMP 38 pin Mictor	HP Logic Analyzer connector
J19	AMP 38 pin Mictor	HP Logic Analyzer connector
J20	AMP 38 pin Mictor	HP Logic Analyzer connector
J21	3 pin 0.1” header	LED connector. If a front-panel mounted LED is to be used (instead of D1) then this should be fitted to pins 3 (cathode) & 2 (anode). Otherwise, to use D1, fit a jumper between pins 1 and 2.

Table 31. Jumpers

Ref	Silk-screen	Pins	Options	Default	Descrip-tion	
JP1	TERM	3	term - off	term	Enables the onboard active SCSI terminations.	
JP2	TPWR	3	tpwr - off	tpwr	Enables power to external SCSI terminations.	
JP3	-	3	33 - 16	33	Switches PCI clocking frequency (both busses) between 33MHz and 16.5MHz.	
JP4	EEWR	2	fit - notfit	notfit	When fitted, enables writing to the EEPROM. Do NOT fit this - it is reserved for production use.	
JP5	-	12	fit - notfit	notfit	Pin 1-2	When not fitted, the Core Card routes JTAG through the CPU only.
				notfit	Pin 3-4	When fitted, allows routing of JTAG (not EJTAG) chain through Atlas/backplane components.
					All other pins are reserved. Do not fit jumpers to these pins.	
JP6	-	8	fit-notfit	->	1-2 & 3-4	ordinary JTAG TDI / TDO connection.
					1-3 & 2-4	Inverts JTAG TDI / TDO connection.
				->	5-6 & 7-8	System slot configuration.
					5-7 & 6-8	Peripheral slot configuration.
JP7	DOWNL	2	fit-notfit	notfit	When fitted, enables cable download of code to FPGA U2. Do NOT fit this - it is reserved for production use.	
JP8	MFWR	2	fit-notfit	notfit	When fitted, allows writing to the Monitor Flash Lock bits from software. It also allows writing to the Monitor Flash itself, regardless of the state of the Lock bits. The disabling function (when not fitted) is overridden by S5-1 when parallel port download is used.	

5.2 Switches

Switches (all on the front panel) are listed below, together with their functions, and default settings. This is how you should receive the Atlas Board. For those switches that are soft-

ware-readable, a switch in position “ON” or “CLOSED” will give a “1” in the appropriate register.

Table 32. Switches

Ref	Type		Default	Description
S1	4-way DIP	S1-1	ON	If ON, system boots from Monitor Flash, else System Flash. Note: Which physical Flash that is being programmed when programming address range 0x1FC0.0000 is also dependent of the position of this switch. When ON, 0x1FC0.0000 maps to the Monitor Flash base address. When OFF, 0x1FC0.0000 maps to the upper 4 Mbytes of System Flash, i.e. 0x1DC0.0000.
		S1-2	ON	If ON, debug serial port is used for debug communication, else the Ethernet port is used for debug. (WinCE convention only)
		S1-3	OFF	If ON, the System Flash is write protected, else it can be written.
		S1-4	OFF	If ON, jump directly to WinCE kernel image on boot, else use Monitor. (WinCE convention only)
S2	8 way DIP		all OFF	This switch provides a value which can be read from the SWITCH register.
S3	Push-button		n/a	Reset button.
S4	Push-button		n/a	NMI/Power ON button. In a benchtop environment, this button will bring the ATX power supply out of standby. It can also be used to generate an NMI to the CPU, for example to shut down the PSU again. This button will also make a hardware shutdown if pressed for more than two seconds. The shutdown is performed when the button is released.
S5	4-way DIP	S5-1	OFF	When ON, enables Flash programming via parallel port.
		S5-2	OFF	When ON, set operation mode to big endian. If the endianess is changed, Atlas must be reset again in order for the new endian mode to take effect. If the board is not reset unpredictable operation can occur.
		S5-3	OFF	No default function.
		S5-4	OFF	When ON, set YAMON in factory default mode e.g. communication on debug serial port is forced to 38.4 kbaud, 8 bits/char, RTS/CTS hardware handshaking and no parity.

DIP switches S1 and S2 are readable by software, and the functions given for S1-2 and S1-4 are conventions applicable in the Windows® CE environment. However S1-1 and S1-3 have direct effects on the function of the hardware.

S5-2, S5-3 and S5-4 are also readable by software, and S5-3 is available for user applications.

For the DIP switches S1, S2 & S5, a dot marks the switch referred to as “1” - this is the left hand one when looking at Atlas on a bench.

5.3 Displays / LEDs

There are three displays near the front panel (see Section 4.1.8 on how to control these) and various LEDs placed on the Atlas board.

Table 33. LEDs on the Atlas Board

Ref	Type	Description
D1	Green LED	ON after reset, is then switched off by the boot software. There is the option to have this replaced by a front-panel mounted LED, connected to J21. J21 must have a jumper fitted between pins 1-2 to enable use of the on-board D1.
D2	8-way bar LED	Function undefined. Can be written to by software.
U1	8 char ASCII display	Used by YAMON to display status during boot, can be used afterwards for any user purpose.
D3-5	Green SMD LEDs	Indicate that power is applied to 3.3V, 5V and 5VSTBY rails respectively.
D6	Green SMD LED	Indicates that CBUS FPGA programming completed OK.
D7	Red SMD LED	Indicates that RSTN is active.
D8	Yellow SMD LED	Indicates TX Ethernet packets.

The two LEDs which are built-in to the RJ45 connector give the status as follows:

LNK/SPD LED - left LED:

Link status	Speed	LED Color
No	X	None/off
OK	10	Yellow
OK	100	Green

ACT/DPLX LED - right LED:

Activity	Duplex	LED Color
None	X	None/off
Yes	Half	Yellow
Yes	Full	Green

Additionally, the yellow LED D8 mounted on the PCB near the connector also gives indication of transmitted packets.

6. Hardware description

This section details the board design details at a reasonable user level. The schematics (Ref[13]) are available if more detailed knowledge is required.

6.1 Compact PCI interface

The Atlas Board is suitable for a “System” slot in a Compact PCI system, and provides the appropriate clock, reset and arbitration functions to the other boards in the rack.

It can be fitted in a peripheral slot, in which case the rack will merely provide power, no CompactPCI function is then supported. Software can detect which slot type Atlas is fitted to by reading the value of the SYSENN signal (1 = system slot) from GPIO[1] on the 21150 bridge chip. If it is identified that the board is not in a System slot, then the clock drivers in the bridge must then be disabled or they will prevent the other devices in the rack from functioning.

The backplane interface is 3.3V/5V compatible, and will therefore function in any rack environment.

6.2 Primary PCI bus

The primary PCI bus is implemented as a 5V, 32bit and 33MHz PCI standard version 2.1 compliant bus that connects the main components on the Atlas Board. The five devices on the primary PCI bus are:

- Core Card connector for connection to the system controller on the Core Card
- Philips SAA9730 IO controller that implements the VGA interface and Video RAM, two USB interfaces, Keyboard and PS2 Mouse interface, I2C interface to serial EEPROM on the Atlas Board and on the Core Card (serial presence detect on the SDRAM module), Serial Debug RS232 Interface and the Ethernet interface.
- Intel 21150AB PCI to PCI bridge that implements the bridge between the PCI bus on the Atlas Board and the Compact PCI interface that connects to the back plane in the rack system
- Symbios SYM53C810A SCSI-2 controller with active termination.
- 5V, 32 bit PCI connector that can be used for debug / trace purposes or for installation of a PCI board.

For configuration purposes the IDSEL and INT# signals to the PCI devices are connected as shown below:

shown below:

Table 34. IDSEL and INT# for PCI devices

Device	IDSEL PCI address line	PCI Interrupts			
		PCI_INTAN	PCI_INTBN	PCI_INTCN	PCI_INTDN
21150 PCI to PCI bridge	PCI_ADP25	none			
SYM53C810A SCSI controller	PCI_ADP26			SCSI_IRQ#	
Core Card	PCI_ADP27	Special interrupt, see Table 4.			
PCI connector	PCI_ADP28	INTA#	INTB#	INTC#	INTD#
SAA9730 IO controller	PCI_ADP29		IO_IRQ#		

6.3 PCI Arbiter

The PCI arbiter controls the request and grant scheduling to the PCI components and is implemented in an Altera MAX7064 EPLD.

The PCI arbiter implements a round-robin scheme where each of the devices have equal priority.

6.4 Power

The board draws power from the Compact PCI backplane, or with 3.3V, 5V and 12V supplied from a standard PC ATX power supply connected to J12. This should comfortably be able to supply enough current for the board and conceivable Core Card options. See Ref [10] for details of a suitable supply.

The -12V rail is only connected to the PCI connector.

6.5 Reset

A push-button on the front panel is provided to reset the board. Alternative sources of reset are:

- The CBUS FPGA when a software-reset register is written.
- A reset push-button fitted to the rack system.
- The power-fail signal FAL# from the CompactPCI backplane. This stops the system before the supply voltage falls too low.
- The EJTAG probe system reset signal.
- An incoming break on “Dbg Ser” UART. This may be disabled by software.

All resets are the same - there is no distinguishing between a “warm” or a “cold” reset.

6.6 Clocks

Both PCI clocks normally run at 33MHz, generated from a 66MHz crystal oscillator plus divider module. Jumper J23 allows the user to decrease this frequency by half.

This will not affect the clock frequency of a CPU mounted on its Core Card. The Core Card generates its own clock.

The different clocks on the board are:

- PCI clock oscillator (66 MHz - divided to 33MHz or 16.5 MHz).
- Ethernet (25 MHz)
- SCSI, CBUS FPGA and Timer (40 MHz)
- Philips IO Controller (27 MHZ)
- Real Time Clock (32 KHz)

6.7 Interrupt

All sources of interrupt on the Atlas Board are handled in an interrupt controller located in the CBUS FPGA. They are combined together to generate interrupts on the Core Card.

All interrupt sources are listed below:

- PCI SERR# signal.
- ATXFAIL from powersupply.
- Real time clock.
- Timer 0 (in CBUS FPGA).
- Four PCI interrupts that are “wired or” signals from the PCI connector and PCI interrupt from Philips SAA9730 IO controller.
- Four Compact PCI interrupts.
- Two interrupts from Core Card.
- ENUM# signal from Compact PCI backplane.
- DEG# signal from Compact PCI backplane.
- Auxiliary Serial port interrupt from the 16550 UART.
- NMI push button.

The NMI interrupt to the Core Card is controlled by the NMI push button, debounced and latched in the CBUS FPGA interrupt controller, see 4.1.2 .

The INTN0 interrupt to the Core Card is controlled by the CBUS FPGA interrupt controller, see Section 4.1.2 , “Interrupt controller.”. INTN[5:1] are always driven inactive.

6.8 Serial ports

There are 2 serial ports on Atlas which are taken out to the front panel to standard, male DB9 connectors (marked “Dbg Ser”, short for “Debug Serial” and “Pd Ser”, short for “Product Serial”).

Each port is electrically identical, with the pinout shown in the table below allowing full hardware handshaking, although one of them (the “Pd Ser” on front panel) is driven by the SAA9730’s main serial port, while the other is driven from a separate 16550 UART chip.

Table 35. Serial port pinouts

PIN NO	Name	Direction
1	CD	Input
2	RXD	Input
3	TXD	Output
4	DTR	Output
5	GND	
6	DSR	Input
7	RTS	Output
8	CTS	Input
9	RI	Input

The signals DSR and RI can be seen on the SAA9730’s GPIO signals as follows:

GPIO[0] - RI
GPIO[1] - DSR

Note that both RI and DSR are active low signals at the GPIO[x] input, meaning that:

Control line ON ~ +12V on RS232 ~ 0V on GPIO[x] ~ “0”
Control line OFF ~ -12V on RS232 ~ 3.3V on GPIO[x] ~ “1”

The following 5-wire symmetric wired cable must be used to guarantee the correct operation of the HW-flow control (with standard PC COM port):

- 2 to 3 (RXD to TXD)
- 3 to 2 (TXD to RXD)
- 5 to 5 (GND to GND)
- 7 to 8 (RTS to CTS)
- 8 to 7 (CTS to RTS)

A recommended data terminal program for PC’s running Windows® OS is Procomm Plus32 from Datastorm Technologies. Basic setup to allow Motorola S-record file download as described below.

Communication settings in the menu area:

- Options->SystemOptions->ModemConnection->System

From here, select the ‘com’ port and click on the ‘Modem/ConnectionProperties’ to set the ‘baudrate’(=38400), ‘parity’(=none), data bits(=8), stop bits(=1) and important select ‘use hardware flow control’.

Download protocol setting in the menu area:

- Options->SystemOptions->ModemConnection->Data:

From here, set ‘current transfer protocol’ to ‘ASCII’, set all delays to ‘0’ and set the CR/LF options to ‘don’t translate CR/LF’. The file to be downloaded to Atlas is selected via the path:

- Data->SendFile

The Serial Debug port can be used to reset the Atlas Board. By default a “Break” condition on the Serial Debug port for more than 10 ms will reset the board exactly as if the reset button had been pressed. This functionality can be disabled or the time can be changed to a different value by programming the BRKRES register in the CBUS FPGA, see Section 4.1.9 .

6.9 Ethernet

The Ethernet supports the 10baseT standard on a twisted pair connection. The interface is implemented in the Philips SAA9730 chip - see Ref [11]. The PHY is a Quality Semiconductor QS6612 10/100BaseTX MII Transceiver for Category 5 Twisted pair cable, which supports auto negotiation of speed (10/100MB/s) and duplex mode (half or full duplex). see Ref [12] This autonegotiation is however disabled, and the interface always runs at 10Mbit/s, half-duplex.

See Section 5.3 for details of the Ethernet LEDs which are built in to the RJ45 connector and the on board LED.

6.10 Video

The video from the SAA9730 is output on a VGA connector on the front panel. Again, consult the manual for the device for all details here. There are 2 Mbytes of DRAM attached to the video controller, which allow for good if not fantastic video resolutions.

6.11 USB / keyboard / mouse / parallel port

All these functions are also provided by the SAA9730. Two USB ports are available on a double connector, the PC keyboard and mouse are on a double mini-DIN connector, both on the front panel. The parallel port is not on the front panel for space reasons - we also assume that most bulk data transfer will be done using Ethernet.

The parallel port also allows the user to reprogram the Flash memory, although this will typically only be done under production. See Section 6.13.

6.12 SCSI

The Narrow SCSI-2 connector J13 is mounted on the furthest edge of the board from the front panel. It is not going to be easy to use this in a rack situation. The SCSI controller is a SYM53C810A (used to be Symbios, now LSI Logic) and unfortunately neither company seem willing to place their documentation on the Web. You may need to order a paper copy by snail-mail if you're going to be doing detailed SCSI work. In this case a SCSI probe, for example the (FuturePlus FS2230/31) may also be very useful.

6.13 Flash memory

Atlas is fitted with 36Mbytes of Flash memory, divided up into two sections, “Monitor” (4 Mbytes) and “System” (32 Mbytes). S1-1 selects which is used for boot i.e. which Flash area that is accessed on address 1FC0.000. Both Monitor Flash and System Flash are always visible. See Section 4 for details of the Atlas Board memory map.

If S1-1 is set to ON, the system will boot from the Monitor Flash. Otherwise, the system will boot from an address in the System Flash.

The System Flash can be reprogrammed by software. There are two protection mechanisms:

- A software addressable register FW_CTRL shall be written with a special code to enable writing to the Flash by asserting its “programming enable” (or equivalent) pin. Writing is disabled by default after reset.
- Switch S1-3 (See Section 5.2) can be set to prevent any writing to the System Flash by always deasserting the programming enable.

The Monitor Flash can also be reprogrammed, but by default, the sectors which contain the YAMON monitor are locked. Lock Bits cannot be cleared unless you fit JP8 (MFWR). This prevents you overwriting YAMON by accident. YAMON stores its environment variables in a writable sector of the Monitor Flash, which is referred to in the YAMON documentation as “Environment Flash”.

Note that while Flash is being reprogrammed by software, the code that performs the programming will have to be copied into RAM and executed there, as the Flash will be inaccessible during this process. The Flash fitted are Intel 64Mbit StrataFlash and FlashFile devices, which are the densest devices available at the time of writing. See Intel’s web site for the documentation, however you may find it easier to use the programming APIs included in YAMON. Note that from a hardware viewpoint, the Flash appears as a 32-bit wide block, with

no individual write control capability to allow writing to just 1 16-bit halfword. However, this function can be achieved through software by running a dummy (e.g. read) cycle on the half-word that is not to be altered.

Both Flash areas can also be reprogrammed by download of an ASCII file to the 1284 parallel port. See Section 8 for details. The YAMON binary distribution contains ASCII files suitable for this download process, which allows upgrade or re-installation of the monitor software.

6.14 EEPROM

The EEPROM is a Fairchild (or alternative) NM24C09 device, with a total of 1024 bytes of memory, which can be write enabled by fitting jumper JP4. Its upper half contains on manufacture the board serial number and the Ethernet MAC address. See Table 2. for details of the memory map.

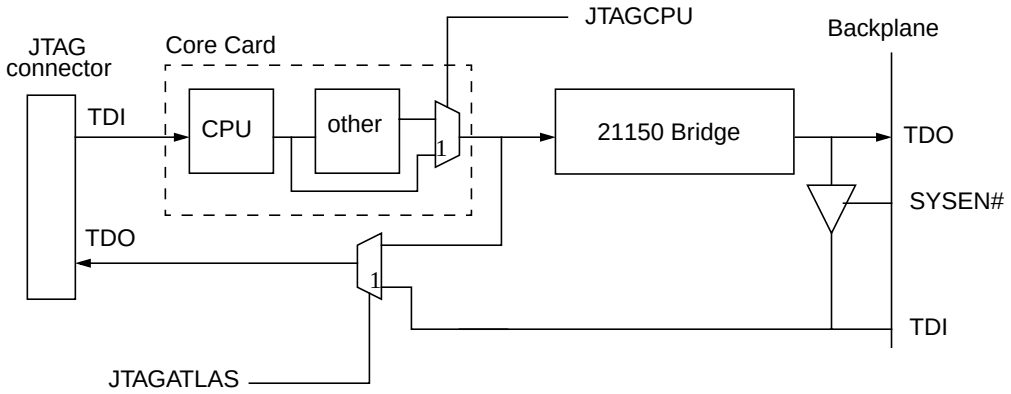
The lower half is unused, and is it possible to use this area. This is NOT recommended, as you risk losing valuable (and supposedly permanent) data.

Note that on some older cards the unused area was accessible without fitting JP4.

6.15 JTAG scan chain.

The Atlas board provides an EJTAG interface for software debugging. The JTAG scan chain on the board is topologically connected as described by Figure 4 below.

Figure 4. JTAG scan chain



The signals JTAGCPU controls the routing of the JTAG chain on the Core Card. JTAGATLAS controls the routing on Atlas. These are both controlled by the setting of jumper J5. Its default setting (no jumpers fitted) is suitable for EJTAG debugging, as the only device then present in the chain is the CPU.

6.16 Debug access

For those long late-night debug sessions, you have access to most if not all interesting signals found on the Atlas board via the debug connectors and test points. See the schematics (Ref[13]) for details.

6.17 Timer/RTC

The Real Time Clock is a Dallas DS1687-5 with internal battery. It contains a timer running at a fixed frequency (32 kHz).

Some Core Cards may need to be able to determine their own clock frequency by software. The RTC timer can be used to do this to sufficient accuracy.

7. SDB Revisions

The following table lists the issued revisions of Atlas and Core Cards that make the released Harp SDB revisions.

Table 36. Harp revisions

Harp SDB revision	Atlas revision	Core card revision	Date released	Description
1.0	02	CPU Card (with QED RM5261 Processor) 00.0	1999/07/05	Initial release for approvals.
1.1	04/05	CoreLV 01.0	2000/03/09	

8. 1284 Flash download format

Both System and Monitor Flash can be programmed and reprogrammed via a download cable that connects directly to a PC parallel port. The CBUS FPGA can read data from this port, and execute the appropriate erase/write cycles in the Flash. The PC must be configured so that its printer port is set to “Generic - text only”, to avoid unpredictable escape sequences being sent. The file format is a sequence of ASCII encoded hex bytes as described below.

How to download:

- Switch OFF both the Atlas, and the PC or workstation that will be used to download.
- Connect the parallel cable between the PCs parallel port, and J14 on the Atlas.
- Switch S5-1 on the Atlas to ON or CLOSED.
- Switch both PC and Atlas ON.
- Run the download script to dump the file to the parallel port on the PC or w/s.
- Disconnect the parallel cable.
- Switch S5-1 to OFF, and reset Atlas.

Note - when S5-1 is set to ON, it overrides any other Flash protection that may have been set by S1-3.

According to the memory map in Section 4. the Monitor Flash is programmed on base address 1E00.0000 and System Flash is programmed on base address 1C00.0000. Which physical Flash that is being programmed when programming address range 1FC0.0000 is dependent of the position of switch S1-1, see Section 5.2 . If any address outside the Flash is addressed the attempt will be ignored.

If S1-1 is ON the Monitor Flash will be programmed on base address 1FC0.0000, and if the switch is OFF it is the upper 4 MBytes from offset 0x01C0.0000 of System Flash that is programmed on base address 1FC0.0000.

Note: When programming the address 1FC0.0010 it is the Flash selected by switch S1-1 that is being programmed, but when reading the address it is overridden and does NOT decode to an address in Flash, but rather to register address REVISION. The only way to read the programmed value back is to read the dedicated System or Monitor Flash address, i.e. address 1C00.0010 or 1E00.0010, see Section 4., “Memory map”

The Flash devices are organised in sectors of 64 Kbyte (Monitor Flash) and 128Kbyte (System Flash). “Erase” and “Set Lock Bit” commands operate on exactly one sector, this being the sector currently addressed. After the last block of 16 words in a sector are written into flash, the address counter has advanced to the next sector. This implies that a Set Address (@) to the sector has to be executed before a Set Lock Bit command (!S) can be issued.

The file to be loaded into the Flash via the 1284 port has the following format:

Type:	ASCII hex (both small and large letters are accepted).
White space:	Any characters below or equal to 20h are ignored, after the character 1Bh (start of a printer initialization command) it ignores any character until next Reset Command.
Word width:	32 bits (data has to be in blocks of 16 words, starting on a 16-word boundary).

A number of codes are used to control code download and Flash memory handling:

Table 37. Download codes

Code	Meaning
@	Sets current writing/erasing address (in Atlas Board physical memory map format). Addresses must be on 16-word boundaries.
!R	Reset download system
!E	Erase the current Flash sector (64 KB)
!C	Clear all Flash lock-bits
!S	Set current Flash sector lock-bit
#	Comment (rest of line)
>	Print command (shows next 8 characters in ASCII display, the command needs exactly 8 non-white space characters). Any character except for "!" and "#" may be printed - use of these in the print command is reserved.
data	data has to be in blocks of 16 words, without interruption of any Comments (#) and Print Commands (>)

Example of code download format:

```
#Example
!R
@1C000000
!E
12345678 23456789 3456789A 456789AB
56789ABC 6789ABCD 789ABCDE 89ABCDEF
9ABCDEF0 ABCDEF01 BCDEF012 CDEF0123
DEF01234 EF012345 F0123456 01234567
# always 16 words in a block
```

After a Reset it will start at 1C00.0000 (i.e. base of System Flash), erase the base sector, and then write the 16 words into offset 0.

If an error should occurs during Flash download the ASCII display will show an error message:

Table 38. Flash download error messages:

Message	**Meaning**
Ill cmd	Illegal command received, e.g. “R” is received (not !R)
Ill !cmd	Illegal “!” command received, e.g. “!A” is received
Ill hex	Illegal hex received (in data or addr), e.g. “ABCDEFGH” both “G” and “H” is illegal characters
Hex exp	Hex expected (always data blocks of 16 words), e.g. a comment (#) is received in the middle of a block of 16 words.
Era susp	Block erase suspended
Err era	Error in block erasure or clear lock-bits
Err prog	Error in programming or set block lock-bits
Low volt	Low programming voltage detected
Lock det	Master lock-bit, Block lock-bit or RP# lock detected

9. Core Card design

This section details the external specification that all Core cards must comply with.

9.1 Interfaces

The following features shall be present on the interface to the Core card.

9.1.1 Power

Power supplies at 3.3V, 5V and 12V are present on the interface. These are positioned such that if a Core card is placed on the Atlas 180 degrees incorrect, all rails are shorted out by a number of pins. This should place the PSU in shutdown mode.

It is **not** guaranteed that the 5V rail will be present before the 3.3V rail.

9.1.2 PCI Bus

The interface to the core card includes a PCI bus. All core cards shall be 5V tolerant on inputs but drive 3.3V on all outputs on the PCI bus.

9.1.3 Clock

The PCI bus clock is driven to the Core card from Atlas. The Core card shall be able to run with this clock at any frequency from 0-33MHz.

9.1.4 Revision number.

The Core card drives a processor-readable revision number, via pins CREV[7:0], down to the Atlas card where the CPU will be able to read them via the CBUS.

This revision number could for example be set via 8 fit/not-fit resistors.

9.1.5 I2C bus

An I2C bus is present on the interface. This will typically be used for interrogating SDRAM DIMMs. See User Manual for the address map.

9.1.6 Interrupts

Six interrupt signals to the MIPS CPU on the Core card are present, as is a single NMI interrupt signal, triggered by a front-panel pushbutton. Core cards must route all of these interrupts to the CPU - if the CPU chip has fewer external interrupt pins, then they should be ORed together.

9.1.7 Endian

The endian control signal BIGEND, driven by Atlas according to setting of S5-2.

9.1.8 CBUS

The “Core bus” is designed to interface to simple devices on the Atlas card which it is necessary that the CPU can access either before the PCI is up and running, or with a low latency (e.g. interrupt controller and Flash).

All core cards shall decode CPU addresses from 0x1C00.0000 to 0x1FFF.FFFF. These address shall then translate into addresses 0x0000.0000 to 0x03FF.FFFF on the CBUS.

Bus signals are:

- CA[25:2] Address bus.
- CD[31:0] Data bus
- CCSN Chip select.
- CWRN Write strobe.
- CRDN Read strobe.

Figure 5. CBUS Read cycle

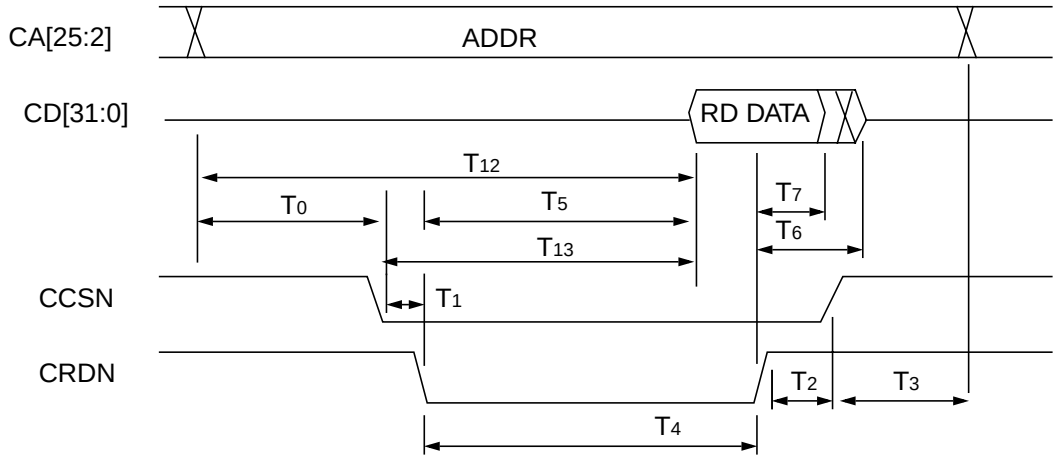


Figure 6. CBUS Write cycle

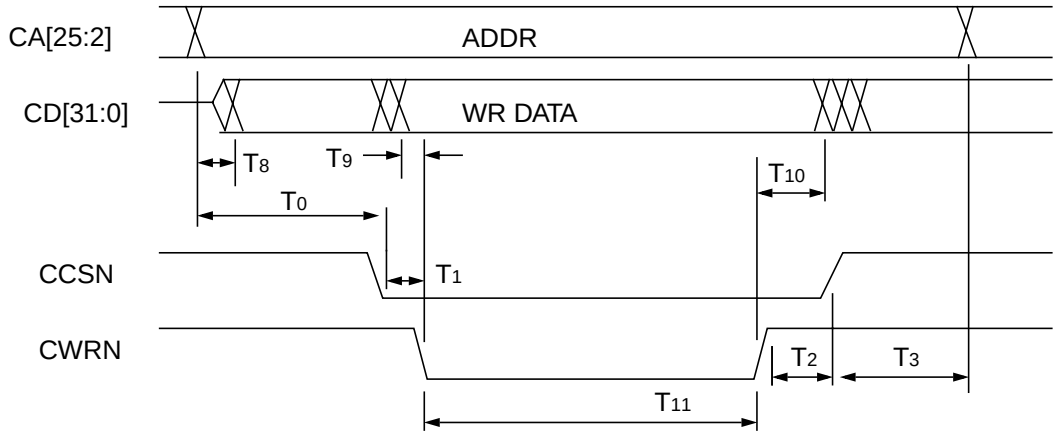


Table 7. CBUS AC timing parameters

Ref	Description	Tmin(ns)	Tmax(ns)
T0	Address valid to CCSN active	5	-
T1	CCSN valid to strobe (CRWN or CRDN) valid	10	-
T2	strobe (CRWN or CRDN) inactive to CCSN inactive	10	-
T3	Address hold from CCSN inactive	20	-
T4	CRDN width	120	-
T5	CRDN active to read data valid	-	120
T6	CRDN inactive to data bus tristated	-	20
T7	Read data hold time after CRDN inactive	0	-
T8	Address valid to data bus driven	0	-
T9	Write data setup to CRWN active	0	-
T10	Write data hold time after CWRN inactive	10	-
T11	CWRN pulse width	75	-
T12	Address valid to read data valid	-	150
T13	CCSN active to read data valid	-	150

9.1.9 EJTAG

The EJTAG signals from the “basic” EJTAG connector are taken to the interface from the front panel connector.

9.1.10 Misc.

Various debug, reserved and presence detect functions - see Section 9.2.

9.2 Signals.

These signals are carried on J3 and J4, which are 200 pin (4-row) 1.27mm pitch connectors, SAMTEC type MOLC-150-31-S-Q (male, fitted to Core card). The interface signal list is as follows:

Table 39. CBUS signals

Signal	Direction (from Core)	Pull (on Core)	Pull (on Atlas)	Function
INTN[5:0]	Input	up	-	Interrupt signals to CPU.
NMIN	Input	up	-	NMI signal to CPU
CD[31:0]	I/O	-	-	CBUS data bus
CCSN	Output	-	-	CBUS chip select
CA[25:2]	Output	-	-	CBUS Address
CRDN	Output	-	-	CBUS read strobe
CWRN	Output	-	-	CBUS write strobe
SCK	I/O	-	-	IIC bus clock
SDA	I/O	-	-	IIC bus data

Signal	Direction (from Core)	Pull (on Core)	Pull (on Atlas)	Function
CINTHIN	Output	-	up	Core card interrupt signal down to Atlas (high priority).
CINTLON	Output	-	up	Core card interrupt signal down to Atlas (low priority).
CREV[7:0]	Output	-	-	Indicates Core card revision as a 6.2 bit binary number
BIGEND	Input	-	-	Sets CPU Endianness.
EJTCK	Input	-	down	
EJTMS	Input	-	up	
EJTRSTN	Input	-	down	
EJTDI	Input	-	up	
EJTDO	Output	-	-	
EJDINT	Input	-	down	
JTGCPU	Input	-	down	Sets Core card's JTAG output to come direct from the CPU rather than also via other circuits.
CGPI[7:0]	Output	-	-	General purpose output from Core which can be read from Atlas.
CGPO[7:0]	Input	-	-	General purpose input to Core which Atlas can drive.
CUU[15:0]	-	-	-	Unused pins, connected to wire-wrap area on both Core and Atlas.
CPRESN	Output	strong down	up	Wired to zero, to indicate presence of Core card.
APRESN	Input	up	strong down	Wired to zero on Atlas, to indicate presence of Atlas.
D12V	Input	-	-	Twelve volt power for possible fan
D5V	Input	-	-	5 Volt power
D3V3	Input	-	-	3.3 Volt power
CPWR_OK	Input	-	-	Indicates that power on both 3V3 and 5V rails is up.
CORE_OK	Output	-	up	Indicates that Core is ready to come out of reset.
RSTN	Input	-	-	Global reset signal.
PCI_AD[31:0]	I/O	-	-	PCI bus
PCI_DEVSELN	I/O	-	-	PCI bus
PCI_CBEN[3:0]	I/O	-	-	PCI bus
PCI_REQN	Output	-	-	PCI bus
PCI_GNTN	Input	-	-	PCI bus
PCI_SERRN	I/O	-	-	PCI bus
PCI_FRAMEN	I/O	-	-	PCI bus
PCI_IRDYN	I/O	-	-	PCI bus
PCI_IDSEL	I/O	-	-	PCI bus

Signal	Direction (from Core)	Pull (on Core)	Pull (on Atlas)	Function
PCI_PAR	I/O	-	-	PCI bus
PCI_STOPN	I/O	-	-	PCI bus
PCI_CLK	Input	-	-	PCI bus
PCI_TRDYN	I/O	-	-	PCI bus
PCI_LOCKN	I/O	-	-	PCI bus
PCI_PERRN	I/O	-	-	PCI bus

9.2.1 J3 Connector

The following pin layout is used for J3.

Table 40. J3 pinout

1	CPWR_OK	51	GND	101	D3V3	151	SCK
2	JTGCPU	52	D5V	102	GND	152	CGPI4
3	CINTHIN	53	GND	103	GND	153	SDA
4	CORE_OK	54	CINTLON	104	GND	154	CGPO7
5	CGPI7	55	GND	105	D3V3	155	CGPO6
6	CD31	56	D5V	106	GND	156	CGPO5
7	CD30	57	GND	107	GND	157	CGPO4
8	CD29	58	GND	108	GND	158	INTN5
9	CD28	59	GND	109	D3V3	159	INTN4
10	CD27	60	D5V	110	GND	160	INTN3
11	CD26	61	GND	111	CGPI0	161	INTN2
12	CD25	62	GND	112	GND	162	INTN1
13	CD24	63	GND	113	D3V3	163	INTN0
14	CD23	64	D5V	114	CA6	164	NMIN
15	CD22	65	GND	115	GND	165	CA10
16	CD21	66	GND	116	GND	166	CA9
17	CD20	67	GND	117	CA5	167	CA8
18	CD19	68	D5V	118	GND	168	CA7
19	CD18	69	GND	119	GND	169	CGPI3
20	CD17	70	GND	120	CA4	170	D12V
21	CD16	71	GND	121	D3V3	171	D12V
22	CD15	72	D5V	122	GND	172	CGPI2
23	CD14	73	GND	123	CA3	173	CPRESN
24	CD13	74	GND	124	GND	174	APRESN
25	CD12	75	GND	125	D3V3	175	BIGEND
26	CD11	76	D5V	126	CA2	176	CGPO3
27	CD10	77	GND	127	GND	177	CGPO2
28	CD9	78	GND	128	GND	178	CGPO1
29	CD8	79	GND	129	CA24	179	CGPO0

Table 40. J3 pinout

30	CD7	80	D5V	130	GND	180	CGPI1
31	CD6	81	GND	131	CA25	181	EJTCK
32	CD5	82	D5V	132	GND	182	GND
33	CD4	83	GND	133	D3V3	183	CREV0
34	CD3	84	GND	134	GND	184	EJTMS
35	CD2	85	GND	135	GND	185	CREV1
36	CD1	86	D5V	136	GND	186	EJTDI
37	CD0	87	GND	137	D3V3	187	CREV2
38	CA23	88	GND	138	GND	188	EJTDO
39	CWRN	89	GND	139	GND	189	CREV3
40	CRDN	90	D5V	140	GND	190	EJTRSTN
41	CCSN	91	GND	141	D3V3	191	CREV4
42	CGPI6	92	GND	142	GND	192	EJDINT
43	CGPI5	93	GND	143	GND	193	CREV5
44	CA22	94	D5V	144	GND	194	CREV6
45	CA21	95	GND	145	D3V3	195	CREV7
46	CA20	96	GND	146	GND	196	CA15
47	CA19	97	GND	147	GND	197	CA14
48	CA18	98	D5V	148	GND	198	CA13
49	CA17	99	GND	149	D3V3	199	CA12
50	CA16	100	CUU4	150	GND	200	GA11

9.2.2 J4 Connector

The following pin layout is used for J4

Table 41. J4 pinout

1	RSTN	51	GND	101	D5V	151	GND
2	PCI_AD31	52	GND	102	GND	152	NC
3	PCI_AD30	53	GND	103	D3V3	153	GND
4	PCI_AD29	54	GND	104	D3V3	154	NC
5	PCI_AD28	55	GND	105	GND	155	GND
6	GND	56	GND	106	D5V	156	NC
7	GND	57	GND	107	GND	157	GND
8	PCI_CLK	58	GND	108	D3V3	158	NC
9	GND	59	GND	109	D3V3	159	GND
10	GND	60	GND	110	GND	160	NC
11	PCI_PAR	61	GND	111	D5V	161	GND
12	PCI_FRAMEN	62	GND	112	GND	162	NC
13	PCI_IRDYN	63	GND	113	D3V3	163	GND

Table 41. J4 pinout

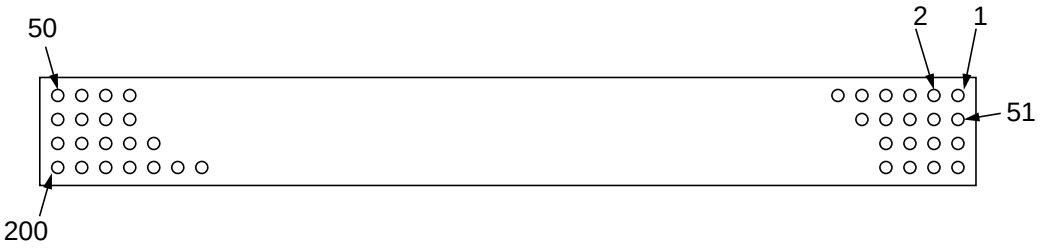
14	PCI_AD27	64	GND	114	D3V3	164	NC
15	PCI_AD26	65	GND	115	GND	165	GND
16	PCI_AD25	66	GND	116	D5V	166	NC
17	PCI_AD24	67	PCI_AD4	117	GND	167	GND
18	PCI_CBEN3	68	GND	118	D3V3	168	NC
19	PCI_CBEN2	69	GND	119	D3V3	169	GND
20	PCI_CBEN1	70	GND	120	GND	170	NC
21	PCI_CBEN0	71	GND	121	D5V	171	GND
22	PCI_AD23	72	PCI_AD3	122	GND	172	NC
23	PCI_AD22	73	GND	123	D3V3	173	GND
24	PCI_AD21	74	GND	124	D3V3	174	NC
25	PCI_AD20	75	GND	125	GND	175	GND
26	PCI_AD19	76	GND	126	D5V	176	NC
27	PCI_AD18	77	PCI_AD2	127	GND	177	GND
28	PCI_AD17	78	GND	128	D3V3	178	NC
29	PCI_AD16	79	GND	129	D3V3	179	GND
30	PCI_TRDYN	80	GND	130	GND	180	NC
31	PCI_STOPN	81	GND	131	D5V	181	GND
32	PCI_LOCKN	82	PCI_AD1	132	GND	182	NC
33	PCI_IDSEL	83	GND	133	D3V3	183	GND
34	PCI_DEVSELN	84	GND	134	D3V3	184	NC
35	PCI_AD15	85	GND	135	GND	185	GND
36	PCI_AD14	86	GND	136	D5V	186	NC
37	PCI_AD13	87	PCI_AD0	137	GND	187	GND
38	PCI_AD12	88	GND	138	D3V3	188	NC
39	PCI_AD11	89	GND	139	D3V3	189	GND
40	PCI_REQN	90	GND	140	GND	190	NC
41	PCI_GNTN	91	GND	141	D5V	191	GND
42	PCI_PERRN	92	GND	142	GND	192	NC
43	PCI_SERRN	93	GND	143	D3V3	193	GND
44	NC	94	GND	144	D3V3	194	NC
45	PCI_AD10	95	GND	145	GND	195	GND
46	PCI_AD9	96	GND	146	D5V	196	NC
47	PCI_AD8	97	GND	147	GND	197	GND
48	PCI_AD7	98	GND	148	D3V3	198	NC
49	PCI_AD6	99	GND	149	D3V3	199	GND
50	PCI_AD5	100	TMS	150	TDO	200	NC

9.3 Physical design

The core card is 170mm x 100mm, and is mounted by pillars at each corner, plus 2 x 200-way (4 row x 25 pin) 1.27mm pitch connectors of type Samtec MOLC-150-31-x-Q. See Figure 9.

Pin numbering on these connectors is as follows:

Figure 8. J3 and J4 pin alignment (view from top of Core card)



Note that one corner pillar (top left in diagram) is placed offset from a symmetrical position, which is to guarantee the board cannot be inserted the wrong way around.

The connectors chosen are low-insertion force ones, however the intention is that the user should be able to “lever” the card up by placing a screwdriver between the mounting pillars and the card. Therefore tracking is forbidden in the areas at the ends of the card.

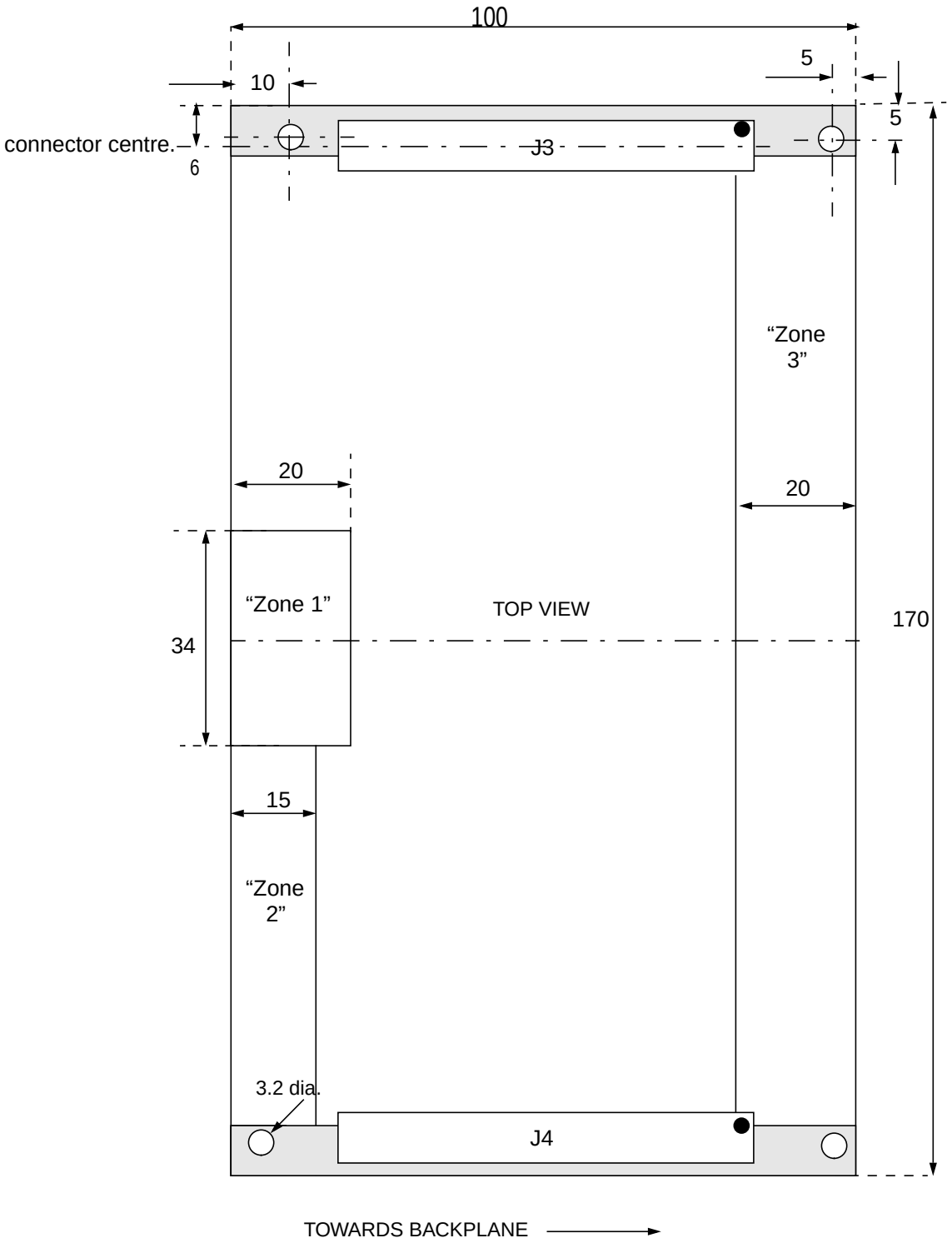
The Harp enclosure provides for a 3-slot wide SDB, therefore the maximum centre-centre shall be 3 x 20.32mm, or 61mm. Allowing for some margin, therefore, the maximum component height above the combined Atlas/Core cards shall be 55mm.

The Core card is mounted at a height of 11mm over the motherboard when using the connectors given above. However, the existing placement of high components on the Atlas and Malta motherboards gives the following restrictions when placing components on the Core board:

Table 42. Core card component height restrictions

Location	Height restrictions on underside
Whole card (default)	No underside SMDs thicker than 6.5mm
Zone 1	No leaded components at all. No underside SMDs thicker than 1.2mm
Zone 2	No underside SMDs thicker than 4.4mm
Zone 3	No underside SMDs thicker than 4.4mm

Figure 9. Core card template layout.



Shaded zone: No tracks on outer layers.

10. References

[1] PCI Local Bus Specification.
Revision 2.1. June 1, 1995

[2] Microsoft® Standard Development Board (SDB) Requirements for Windows® CE
Version 5.4.

[3] CompactPCI Specification
Version 2.0 R2.1

[4] Microsoft Harp enclosure requirements
Version 1.0

[5] MIPS YAMON™ User's Manual
MD00008

[6] MIPS YAMON™ Reference Manual
MD00009

[7] EJTAG specification.
Version 2.5

[8] Dallas DS1687 data sheet.
Available from http://www.dalsemi.com

[9] Texas Instruments TI 16C550C datasheet.
Available from http://www.ti.com

[10] Intel ATX Power Supply Design Guide.
Version 0.9

[11] Philips SAA9730 Data sheet.

[12] Quality Semiconductor QS6612 data sheet.
Available from http://www.qualitysemi.com

[13] MIPS Atlas™ Schematic
MD00019

11. Revision history

Table 43. Revision history

Rev.	Date	Comments
01.00	1999/12/15	Initial release.
01.01	2000/01/03	Updated with Core card specification.
01.02	2000/01/13	Slight modifications to CBUS timing - CRDN pulse shorter, T12 added. Improved board layout diagram (Section 5). Added Front panel layout Fixed erroneous table and figure numbering
01.03	2000/02/07	Fixed copyright date Made CLOSED an alias for ON w.r.t. switch settings. Fixed TOC references
01.04	2000/03/22	Removed Header Files section, replaced with reference to YAMON Ref. Manual and source deliverables. Allowed 5-wire serial cable. Added details of Flash programming and protection, and YAMON upgrading. Added EJTAG routing. Added PCI arbiter description. Required all INT pins to be connected on Core Card. STATUS register MFWR field indicates write enabled. Removed Atlas & Core revisions table. Moved to Errata sheet. Added full name for CPU Card in SDB revisions table. Minor fixes throughout.
01.05	2000/04/13	Fixed incorrect references to Pd/Dbg Ser ports.
01.06	2000/06/20	Updated CBUS timing with CCSN to read data valid timing. Updated copyright notice.
01.07	2000/06/30	Updated Getting Started section. Renamed bitfields in Interrupt controller. Fixed definition of MFWR in STATUS register. Minor edits.
01.08	2000/07/10	Added more detail to effect of S1 on memory mapping.
01.09	2000/08/30	Added component height restrictions to Core card specification. Added details to memory map including restriction of boot flash area to 1MB. Updated definition of JP8. Added reservation of ! and # in 1284 flash download format description.
01.10	2000/10/12	Whole of EEPROM now write protected by JP4.